What is claimed is:

1. A method of handling operation in a multi-threaded processing system, comprising:

determining if a stalled operation of a first thread is due to a loading of data from a memory device; and

flushing an instruction from said first thread from a pipeline of said processing system when data is to be loaded from said memory device before executing said instruction.

2. The method of claim 1 wherein said memory device is system memory coupled to a memory bus.

3. The method of claim 1 further comprising:

marking said instruction as a miss.

4. The method of claim 3 further comprising:

rescheduling said instruction to be executed in said pipeline.

5. A method of handling operation in a multi-threaded processing system, comprising:

determining if a stalled operation of a first thread is due to a loading of data from a memory device; and

flushing an instruction from said first thread from a pipeline of said processing system when data is to be loaded after a predetermined number of clock cycles from said memory device before said instruction can be executed.

6. The method of claim 5 wherein said memory device is system memory coupled to a memory bus.

7. The method of claim 6 further comprising:

marking said instruction as a miss.

8. The method of claim 7 further comprising:

rescheduling said instruction to be executed in said pipeline.

9. The method of claim 8 further comprising:

executing said instruction when data is loaded from said memory device.

10. A processing system comprising:

a scheduler to pass instructions from first thread and second threads to an execution pipeline; and

pipeline control logic coupled to said execution pipeline to determine if a stalled execution of a first thread is due to a loading of data from a memory device and to flush an instruction from said first thread from said execution pipeline when data is to be loaded from said memory device before said instruction can be executed.

11. The processing system of claim 10 wherein said pipeline control logic is to mark said instruction as a miss.

12. The processing system of claim 10 further comprising:

an exception and retirement logic coupled to said execution pipeline.

13. The processing system of claim 12 wherein said instruction marked as a miss is to be detected by said exception and retirement logic.

14. The processing system of claim 13 further comprising:

a fetch unit to provide said instruction to said scheduler.

15. The processing system of claim 14 wherein said pipeline control logic is to cause said instruction to be executed when data is loaded from said memory device.

16. A computing system comprising:

a memory bus coupled to system memory; and

a processing system coupled to said memory bus, said processing system including

a scheduler to pass instructions from first thread and second threads to an execution pipeline; and

pipeline control logic coupled to said execution pipeline to determine if a stalled execution of a first thread is due to a loading of data from system memory and to flush an instruction from said first thread from said execution pipeline when data is to be loaded from said system memory before said instruction can be executed.

17. The computing system of claim 16 wherein said pipeline control logic is to mark said instruction as a miss.

18. The computing system of claim 10 wherein said processing system further includes an exception and retirement logic coupled to said execution pipeline.

19. The computing system of claim 18 wherein said instruction marked as a miss is to be detected by said exception and retirement logic.

20. The computing system of claim 19 wherein said processing system further includes a fetch unit to provide said instruction to said scheduler.

21. The computing system of claim 20 wherein said pipeline control logic is to cause said instruction to be executed when data is loaded from said system memory.